

PPPPPPPPPPPP		AAAAAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTT		LLL
PPPPPPPPPPPP		AAAAAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTT		LLL
PPPPPPPPPPPP		AAAAAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPPPPPPPPPPP		AAA	AAA	SSSSSSSSSS		RRRRRRRRRRRR		TTT		LLL
PPPPPPPPPPPP		AAA	AAA	SSSSSSSSSS		RRRRRRRRRRRR		TTT		LLL
PPPPPPPPPPPP		AAA	AAA	SSSSSSSSSS		RRRRRRRRRRRR		TTT		LLL
PPP		AAAAAAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
PPP		AAAAAAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
PPP		AAAAAAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA		SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA		SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA		SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA		SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL

52

Sym

PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PAC

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PAS

PAS
PAS

PAGE

PAS

RAC

PAS

PAS

DAS

PAS
PAS

PAS

PAS

PAS

PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS

PAS

```
PPPPPPPP      AAAAAA      SSSSSSSS      VV      VV      MM      MM
PPPPPPPP      AAAAAA      SSSSSSSS      VV      VV      MM      MM
PP      PP      AA      AA      SS      VV      VV      MMM      MMM
PP      PP      AA      AA      SS      VV      VV      MMM      MMM
PP      PP      AA      AA      SS      VV      VV      MM      MM
PP      PP      AA      AA      SS      VV      VV      MM      MM
PPPPPPPP      AA      AA      SSSSSS      VV      VV      MM      MM
PPPPPPPP      AA      AA      SSSSSS      VV      VV      MM      MM
PP      AAAAAAAAAA      SS      VV      VV      MM      MM
PP      AAAAAAAAAA      SS      VV      VV      MM      MM
PP      AA      AA      SS      VV      VV      MM      MM
PP      AA      AA      SS      VV      VV      MM      MM
PP      AA      AA      SSSSSSSS      VV      VV      MM      MM
PP      AA      AA      SSSSSSSS      VV      VV      MM      MM
                                     ....
                                     ....
                                     ....
                                     ....
```

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```

```
0001 0 MODULE PAS$$VM ( %TITLE 'Allocate/deallocate virtual memory'
0002 0 IDENT = '1-001' ! File: PASVM.B32 Edit: SBL1001
0003 0 ) =
0004 1 BEGIN
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *
0027 1 *****
0028 1
0029 1
0030 1 ++
0031 1 FACILITY: Pascal Language Support
0032 1
0033 1 ABSTRACT:
0034 1
0035 1 This module contains procedures to allocate and deallocate
0036 1 virtual memory.
0037 1
0038 1 ENVIRONMENT: User mode - AST reentrant
0039 1
0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. SBL 1-April-1981
0045 1 --
0046 1
```


PASS\$VM
1-001

Allocate/deallocate virtual memory
Declarations

L 10
16-Sep-1984 02:13:42
14-Sep-1984 12:52:00

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PASRTL.SRC]PASVM.B32;1 Page 2
(2)

```

: 48 0047 1 %SBTTL 'Declarations'
: 49 0048 1
: 50 0049 1 PROLOGUE DEFINITIONS:
: 51 0050 1
: 52 0051 1
: 53 0052 1 REQUIRE 'RTLIN:PASPROLOG'; ! Linkages, externals, PSECTs, structures
: 54 0116 1
: 55 0117 1
: 56 0118 1 TABLE OF CONTENTS:
: 57 0119 1
: 58 0120 1
: 59 0121 1 FORWARD ROUTINE
: 60 0122 1 PASS$GET_VM: CALL VM; ! Allocate virtual memory
: 61 0123 1 PASS$FREE_VM: NOVALUE; ! Deallocate virtual memory
: 62 0124 1
: 63 0125 1
: 64 0126 1 MACROS:
: 65 0127 1
: 66 0128 1 NONE
: 67 0129 1
: 68 0130 1 EQUATED SYMBOLS:
: 69 0131 1
: 70 0132 1 NONE
: 71 0133 1
: 72 0134 1 FIELDS:
: 73 0135 1
: 74 0136 1 NONE
: 75 0137 1
: 76 0138 1 OWN STORAGE:
: 77 0139 1
: 78 0140 1 NONE
```

```

80 0141 1 %SBTTL 'PAS$$GET_VM - Allocate virtual memory'
81 0142 1 GLOBAL ROUTINE PAS$$GET_VM (
82 0143 1     PFV: REF $PAS$PFV_FILE_VARIABLE,      ! File variable
83 0144 1     LENGTH                                ! Length to allocate
84 0145 1 ): CALL_VM =
85 0146 1
86 0147 1
87 0148 1 ++
88 0149 1 FUNCTIONAL DESCRIPTION:
89 0150 1     This procedure allocates a block of virtual memory and returns
90 0151 1     a pointer to the block as its function value. The memory allocated
91 0152 1     is zero-filled.
92 0153 1
93 0154 1 CALLING SEQUENCE:
94 0155 1
95 0156 1     block_addr.wa.v = PAS$$GET_VM (PFV.rr.r, LENGTH.rl.v)
96 0157 1
97 0158 1 FORMAL PARAMETERS:
98 0159 1
99 0160 1     PFV                - The Pascal File Variable (PFV) passed by reference.
100 0161 1                    The structure of the PFV is defined in PASPFV.REQ.
101 0162 1
102 0163 1     LENGTH            - The length in bytes of the block to allocate.
103 0164 1                    Must not be greater than 65535.
104 0165 1
105 0166 1 IMPLICIT INPUTS:
106 0167 1
107 0168 1     NONE
108 0169 1
109 0170 1 IMPLICIT OUTPUTS:
110 0171 1
111 0172 1     NONE
112 0173 1
113 0174 1 ROUTINE VALUE:
114 0175 1
115 0176 1     A pointer to the allocated block.
116 0177 1
117 0178 1 SIDE EFFECTS:
118 0179 1
119 0180 1     Allocates virtual memory.
120 0181 1
121 0182 1 SIGNALLED ERRORS:
122 0183 1
123 0184 1     INSVIRMEM - Insufficient virtual memory
124 0185 1
125 0186 1 --
126 0187 1
127 0188 2 BEGIN
128 0189 2
129 0190 2 LOCAL
130 0191 2     BLOCK_ADDR;                ! Address of allocated block
131 0192 2
132 0193 2 IF .LENGTH GTRU 65535
133 0194 2 THEN
134 0195 2     $PAS$BUGCHECK (BUG_BADVMSIZE);
135 0196 2
136 0197 2 IF NOT LIB$GET_VM (LENGTH, BLOCK_ADDR)
```

PASS\$VM
1-001

Allocate/deallocate virtual memory
PASS\$GET_VM - Allocate virtual memory

N 10
16-Sep-1984 02:13:42
14-Sep-1984 12:52:00

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PASRTL.SRC]PASVM.B32;1
Page 4
(3)

```
: 137      0198  2      THEN
: 138      0199      $PASS$IO_ERROR (PASS$_INSVIRMEM,0);
: 139      0200
: 140      0201      CH$FILL (0, .LENGTH, .BLOCK_ADDR); ! Zero-fill
: 141      0202
: 142      0203      RETURN .BLOCK_ADDR;
: 143      0204
: 144      0205  1      END;
```

! End of routine PASS\$GET_VM

.TITLE PASS\$VM Allocate/deallocate virtual memory
.IDENT \1-001\

.EXTRN PASS\$GET_VM, PASS\$FREE_VM
.EXTRN PASS\$BUGCHECK, LIB\$GET_VM
.EXTRN PASS\$SIGNAL, PASS\$_INSVIRMEM

.PSECT _PASS\$CODE,NOWRT, SHR, PIC,2

.ENTRY PASS\$GET_VM, Save R2,R3,R4,R5

```
SUBL2  #4, SP      : 0142
CMPL   LENGTH, #65535 : 0193
BLEQU  1$
PUSHL  #4          : 0195
CALLS  #1, PASS$BUGCHECK
BRB    3$
PUSHL  SP          : 0197
PUSHAB LENGTH
CALLS  #2, LIB$GET_VM
BLBS   R0, 2$
CLRL   -(SP)       : 0199
MOVZBL #PASS$_INSVIRMEM, -(SP)
CALLS  #2, PASS$SIGNAL
BRB    3$
MOVC5  #0, (SP), #0, LENGTH, @BLOCK_ADDR : 0201
MOVL   BLOCK_ADDR, R0 : 0203
RET
CLRL   R0          : 0205
RET
```

; Routine Size: 71 bytes, Routine Base: _PASS\$CODE + 0000

```
: 145      0206  1
: 146      0207  1 !<BLF/PAGE>
```



```

148 0208 1 %SBTTL 'PASS$FREE_VM - Deallocate virtual memory'
149 0209 1 GLOBAL ROUTINE PASS$FREE_VM (
150 0210 1     LENGTH,                                ! Length to allocate
151 0211 1     BLOCK_ADDR: REF VECTOR [, LONG]    ! Address of block to deallocate
152 0212 1     ): NOVALUE =
153 0213 1
154 0214 1 ++
155 0215 1 FUNCTIONAL DESCRIPTION:
156 0216 1
157 0217 1     This procedure deallocates a block of virtual memory.
158 0218 1
159 0219 1 CALLING SEQUENCE:
160 0220 1
161 0221 1     CALL PASS$FREE_VM (LENGTH.rl.v, BLOCK_ADDR.ma.r)
162 0222 1
163 0223 1 FORMAL PARAMETERS:
164 0224 1
165 0225 1     LENGTH                - The length in bytes of the block to deallocate.
166 0226 1
167 0227 1     BLOCK_ADDR            - The address of the block to deallocate
168 0228 1
169 0229 1 IMPLICIT INPUTS:
170 0230 1
171 0231 1     NONE
172 0232 1
173 0233 1 IMPLICIT OUTPUTS:
174 0234 1
175 0235 1     NONE
176 0236 1
177 0237 1 ROUTINE VALUE:
178 0238 1
179 0239 1     NONE
180 0240 1
181 0241 1 SIDE EFFECTS:
182 0242 1
183 0243 1     Deallocates virtual memory.
184 0244 1
185 0245 1 SIGNALLED ERRORS:
186 0246 1
187 0247 1     BUGCHECK
188 0248 1
189 0249 1 --
190 0250 1
191 0251 2 BEGIN
192 0252 2
193 0253 2 IF NOT LIB$FREE_VM (LENGTH, BLOCK_ADDR [0])
194 0254 2 THEN
195 0255 2     $PASS$BUGCHECK (BUG_FREEVMFAIL);
196 0256 2
197 0257 2 BLOCK_ADDR [0] = 0;          ! Invalidate pointer
198 0258 2
199 0259 2 RETURN;
200 0260 2
201 0261 1 END;

```

PASS\$VM
1-001

Allocate/deallocate virtual memory
PASS\$FREE_VM - Deallocate virtual memory

C 11
16-Sep-1984 02:13:42
14-Sep-1984 12:52:00

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PASRTL.SRC]PASVM.B32;1
Page 6 (4)

```
00000000G 00
0A
00000000G 00
08 AC DD 00002
04 AC 9F 00005
02 FB 00008
50 E8 0000F
02 DD 00012
01 FB 00014
04 0001B
08 BC D4 0001C 1$:
04 0001F
```

```
.EXTRN LIB$FREE_VM
.ENTRY PASS$FREE_VM, Save nothing
PUSHL BLOCK_ADDR
PUSHAB LENGTH
CALLS #2, LIB$FREE_VM
BLBS R0, 1$
PUSHL #2
CALLS #1, PASS$BUGCHECK
RET
CLRL @BLOCK_ADDR
RET
```

0209
0253
0255
0257
0261

; Routine Size: 32 bytes, Routine Base: _PASS\$CODE + 0047

: 202 0262 1
: 203 0263 1 !<BLF/PAGE>

PASS\$VM
1-001 Allocate/deallocate virtual memory
PASS\$FREE_VM - Deallocate virtual memory

D 11
16-Sep-1984 02:13:42 VAX-11 Bliss-32 V4.0-742 Page 7
14-Sep-1984 12:52:00 DISK\$VMSMASTER:[PASRTL.SRC]PASVM.B32;1 (5)

: 205 0264 1 END
: 206 0265 1
: 207 0266 0 ELUDOM

! End of module PASS\$VM

PSECT SUMMARY

: Name Bytes Attributes
: _PASS\$CODE 103 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	0	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	33	7	33	00:00.4

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASVM/OBJ=OBJ\$:PASVM MSRC\$:PASVM/UPDATE=(ENH\$:PASVM)

: Size: 103 code + 0 data bytes
: Run Time: 00:04.4
: Elapsed Time: 00:16.9
: Lines/CPU Min: 3668
: Lexemes/CPU-Min: 7586
: Memory Used: 47 pages
: Compilation Complete

0297

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY